

R. Aubry · S.R. Idelsohn · E. Oñate

# Fractional step like schemes for free surface problems with thermal coupling using the Lagrangian PFEM

Received: 29 December 2005 / Accepted: 10 February 2006 / Published online: 25 April 2006  
© Springer-Verlag 2006

**Abstract** The method presented in Aubry et al. (Comput Struc 83:1459–1475, 2005) for the solution of an incompressible viscous fluid flow with heat transfer using a fully Lagrangian description of motion is extended to three dimensions (3D) with particular emphasis on mass conservation. A modified fractional step (FS) based on the pressure Schur complement (Turek 1999), and related to the class of algebraic splittings Quarteroni et al. (Comput Methods Appl Mech Eng 188:505–526, 2000), is used and a new advantage of the splittings of the equations compared with the classical FS is highlighted for free surface problems. The temperature is semi-coupled with the displacement, which is the main variable in a Lagrangian description. Comparisons for various mesh Reynolds numbers are performed with the classical FS, an algebraic splitting and a monolithic solution, in order to illustrate the behaviour of the Uzawa operator and the mass conservation. As the classical fractional step is equivalent to one iteration of the Uzawa algorithm performed with a standard Laplacian as a preconditioner, it will behave well only in a Reynold mesh number domain where the preconditioner is efficient. Numerical results are provided to assess the superiority of the modified algebraic splitting to the classical FS.

**Keywords** Lagrangian description · Mixed incompressible element · Coupled thermo mechanical analysis · Pressure schur complement · Generalized stokes solver

## 1 Introduction

The Navier–Stokes equations have been traditionally associated with an Eulerian description of motion, where the veloc-

ity is known on each spatial point of the problem domain. The Lagrangian formulation offers a different point of view, as each particle knows its velocity. At the continuum level, both descriptions are strictly equivalent but give rise to different implementations and difficulties. A major drawback with a Lagrangian approach is perhaps the necessity to remesh frequently, which was not affordable until very recently. As mesh generation has undergone amazing progresses these last twenty years [31, 32, 40, 41, 63], this has made the use of the Lagrangian formulation now possible. At the computational level, a Lagrangian approach offers many advantages, as no convective term appears in the time derivative which means, amongst other things:

- no stabilization of the convective term is necessary [38].
- the matrices to be solved are symmetric which provides minimization properties with short term recurrences for iterative solvers [3, 56].
- it almost halves storage as matrices are symmetric.
- optimal preconditioners are available for the generalized Stokes problem but not for the Navier-Stokes equations [17, 29, 43].
- for free surface problems, it provides an explicit description of the free surface and no additional transport or reinitialization equations (Level set [46], VOF [36], pseudo concentration) need to be solved. It is furthermore far less diffusive [46] in this context. Finally, the solution of the non linear problem is the final solution whereas the solution with a level set-like method is most of the time explicitly convected due to instabilities with elements that are fluid then gas inside the non linear process.
- only the domain filled by the fluid is meshed.
- boundary conditions (temperature, pressure, heat flux) on free surfaces are straightforward to impose.

In this paper, the particle finite element method (PFEM) [26, 39, 45] (E. Oñate et al. 2004, submitted) is used with various mixed elements in order to solve thermal convection for incompressible fluid flows. Mass conservation constitutes an important problem in fluid mechanics. As depicted in [48], mass conservation is usually only partially weakly verified.

R. Aubry (✉) · E. Oñate  
Universidad Politécnica de Cataluña,  
International Center for Numerical Methods in Engineering (CIMNE),  
Barcelona, Spain  
E-mail: romain@cimne.upc.edu  
E-mail: onate@cimne.upc.edu

S.R. Idelsohn  
ICREA Research Professor at CIMNE  
E-mail: sergio@cimne.upc.edu

In a Lagrangian description with free surfaces, the mass loss is immediately observed as the volume of the fluid changes during the computation, providing meaningless results. However, it should be pointed out that this phenomenon happens whatever description has been chosen, the Lagrangian description bringing it only more obvious.

A very common approach to solve the Navier–Stokes equations is the use of the FS method, performed either at the continuous [20, 60] or at the algebraic [15, 22, 35, 54, 57, 61] level. The main advantages are the uncoupling between velocity and pressure, and possibly for each component of the velocity but, by far the most significant is the circumvention of the saddle point problem due to incompressibility. It offers a very cheap and accurate alternative to the monolithic resolution under appropriate mesh Reynolds number domain, as seen below. However, it will be shown that the classical FS is responsible for the mass loss because the pressure, which does not need to be prescribed at the free surface in a monolithic approach, as it is taken into account by the variational formulation as natural boundary condition, must be strongly imposed due to the presence of the pressure Laplacian. This drawback will be circumvented by a modified algebraic splitting (MAS) [57] based on the pressure Schur complement (PSC). The different relationships between time discretization, mixed elements and the choice of the solver are stressed to highlight mass conservation.

Here, the work of [5] is extended to 3D problems and important improvements have been realized in the flow solver. Changes have been performed to ensure a very accurate mass conservation, at least from a divergence point of view. A short review of the Navier–Stokes and heat transport Lagrangian equations is then presented. The discretization of these equations in space performed by a mixed FEM method and in time by a  $\theta$ -scheme is detailed. The important choice of fluid solver is discussed. The solution of the thermal problem is dealt with next. Finally, numerical results are presented.

## 2 The Navier–Stokes and the heat equations in the Lagrangian formulation

In this section, the main features of the Lagrangian approach are recalled. As usual, unknowns of interest, here velocity, displacement, pressure and temperature, will be written with capital and lower case letters, in the reference and current configuration respectively. For more details, see [5].

### 2.1 The heat equation

In a Lagrangian description, the heat equation reads:

$$\rho_0 C \frac{DT}{Dt} = \text{div}_X (\kappa J \mathbf{F}^{-1} \mathbf{F}^{-T} \text{grad}_X(T)) + \mathbf{f}_T, \quad (1)$$

where  $T$  is the temperature,  $\kappa$  the thermal conductivity,  $\rho_0$  the density,  $C$  the heat capacity,  $\mathbf{F}^{-1}$  represents the transformation gradient,  $J$  its determinant, the jacobian, and  $\mathbf{f}_T$  possible source terms contributions. The problem must be completed

with standard boundary conditions including heat convection and radiation fluxes, and imposed temperature boundaries. Another way to write (1) is to write the temperature and the heat flux with two different variables:

$$\rho_0 C \frac{DT}{Dt} = -\text{div}_X (J \mathbf{F}^{-1} \mathbf{q}), \quad (2)$$

$$\mathbf{q} = -\kappa \mathbf{F}^{-T} \text{grad}_X(T). \quad (3)$$

The system of equations (2) and (3) is the basis to a mixed approach for parabolic problems discussed later.

### 2.2 The Navier–Stokes equations

The Lagrangian equations of motion for a Newtonian incompressible fluid are:

$$\rho_0 \frac{D\mathbf{V}}{Dt} = \rho_0 \mathbf{f}_V - \text{div}_X (J p \mathbf{F}^{-T}) + \text{div}_X (\mu J (\text{grad}_X(\mathbf{V}) \mathbf{F}^{-1} + \mathbf{F}^{-T} \text{grad}_X^T(\mathbf{V})) \mathbf{F}^{-T}) \quad (4)$$

$$J = 1 \quad \text{or} \quad \text{Tr}(\text{grad}_X(\mathbf{V}) \mathbf{F}^{-1}) = 0 \quad (5)$$

where  $\text{Tr}$  is the trace operator,  $\mathbf{V}$  the velocity,  $\mu$  the dynamic viscosity and  $\mathbf{f}_V$  an external force term. Here also, standard appropriate boundary conditions must be added for a well defined problem.

*Remark 1* The second part of Eq. (5) is the time derivative of the first part so that, as noted in [9], it is less accurate from a numerical point of view, whereas both formulae are strictly equivalent at the continuum level. For incompressible finite strain analysis, the first part of Eq. (5) is weakly imposed. However, as noted in [5], the constitutive equation for an hyperelastic material is given in the reference configuration and it is a version with the time derivative, which gives an awkward linearisation. Numerical results for both the Jacobian and the norm of the divergence will be shown in the last section.

## 3 Discretization of the equations

In this section, the weak form of the equations introduced in the previous section are discretized in space and time, and the finite elements considered in this work are presented.

### 3.1 The algebraic system

A classical  $\theta$ -scheme [23] is used to march in time. For the spatial discretization, a standard mixed finite element method [24] is performed so that the system reads, with  $\mathbf{U}$  the displacement and  $\mathbf{V}$  the velocity:

$$\begin{aligned} \frac{\rho_0}{\delta t} \mathbf{M}_V (\mathbf{V}^{n+1} - \mathbf{V}^n) + \mu \mathbf{K}_V (\mathbf{U}^{n+\theta}) \mathbf{V}^{n+\theta}, \\ + \mathbf{B}^T (\mathbf{U}^{n+\theta}) \mathbf{P}^{n+1} = \mathbf{F}_V^{n+\theta}, \end{aligned} \quad (6)$$

$$\mathbf{B} (\mathbf{U}^{n+\theta}) \mathbf{V}^{n+1} = 0, \quad (7)$$

where the previous matrices are:

$$\mathbf{M}_{Vij}^{ab} = \delta_{ab} \int_{\Omega_0} \mathbf{N}_i \mathbf{N}_j dV_0, \quad (8)$$

$$\mathbf{K}_{Vij} = \int_{\Omega_0} J \left( \mathbf{grad}_X(\mathbf{N}_i) \mathbf{F}^{-1} + \mathbf{grad}_X^T(\mathbf{N}_i) \mathbf{F}^{-T} \right) \cdot \mathbf{grad}_X(\mathbf{N}_j) \mathbf{F}^{-1} dV_0, \quad (9)$$

$$\mathbf{B}_{ij} = - \int_{\Omega_0} \Psi_i \text{Tr}(\mathbf{grad}_X(\mathbf{N}_j) \mathbf{F}^{-1}) J dV_0, \quad (10)$$

where the subscripts refer to the node indices and the superscripts to the space indices, the  $\mathbf{N}_i$  and the  $\Psi_i$  are the velocity and pressure shape functions. The non-linear dependence of the matrices to the displacement variable has been emphasized to denote the Lagrangian description of the discretization. Writing the system (6) and (7) as a whole system, and including all the known terms in the right hand side, the algebraic system reads:

$$\begin{pmatrix} \mathbf{A} & \mathbf{B}^T \\ \mathbf{B} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{V} \\ \mathbf{P} \end{pmatrix} = \begin{pmatrix} \tilde{\mathbf{F}}_V \\ \mathbf{0} \end{pmatrix}. \quad (11)$$

Here, (11) represents a classical generalized Stokes operator as there is no convective term and  $\mathbf{A}$  has the form:

$$\mathbf{A} = \frac{\rho_0}{\delta t} \mathbf{M}_V + \theta \mu \mathbf{K}_V. \quad (12)$$

For the temperature, the algebraic system reads:

$$\left( \rho_0 C \frac{\mathbf{M}_T}{\delta t} + \theta \kappa \mathbf{K}_T(\mathbf{U}) \right) \mathbf{T} = \tilde{\mathbf{F}}_T, \quad (13)$$

with appropriate right hand side containing source and boundary terms, and temperature terms at previous time steps. Matrices  $\mathbf{M}_T$  and  $\mathbf{K}_T$  are the counterparts of the previous matrices  $\mathbf{M}_V$  and  $\mathbf{K}_V$  for a scalar unknown. For sake of clarity, all the physical properties have been considered as constant but the extension to variable physical properties is straightforward.

### 3.2 Compatible pressure/velocity spaces

System (6) and (7) requires the satisfaction of the classical LBB condition to be invertible [18, 55]. In this work, three kinds of elements have been considered:

- a continuous  $P1/P1$  element which does not satisfy the inf-sup but is commonly used with a first-order FS, giving some stability properties [22].
- a continuous  $P1 + /P1$  element belonging to the bubble family [1], where the bubble has been discretized by four subtetrahedra, also called the “hat-function”, allowing an exact integration [49, 50].
- a  $P1 + +/P1$  element with bubble and linear discontinuous pressure [14].

The last two elements are div-stable using classical projection operators for continuous and discontinuous pressures [18].

**Remark 2** Equivalence between the bubble element and stabilized formulation has been proved [7, 8, 51, 52, 58]. Actually, the bubble gives the local Schur complement  $\mathbf{B}\mathbf{A}^{-1}\mathbf{B}^T$  which, in case of the Stokes problem is equivalent to a scaled Laplacian operator as the bubble is local to an element. However, the bubble for the  $P1 + +/P1$  element gives stability for the linear part of the pressure only, the element being already stable for  $P_0$  pressure. Furthermore, if the static condensation of the bubble in the mini element gives rise to a symmetric positive definite (SPD) block in  $\mathbf{A}$  (up to the hydrostatic pressure mode), it is not the same by condensing the bubble and the linear part of the pressure of the  $P1 + +/P1$ . This block in  $\mathbf{A}$  is then indefinite, reflecting the coupling between pressure and velocity. This result was verified by computing the eigenvalues of this block in  $\mathbf{A}$  after condensation.

## 4 The generalized Stokes solver

The previous equations are non linear in the displacement variable as the final domain is not known. After Picard linearization, a generalized Stokes problem needs to be solved at each non linear iteration. This section presents the solver used in this work, by highlighting a variant of the Uzawa method, and its relationships with the algebraic splitting methods.

### 4.1 Approximation of the PSC

The main difficulty of the Stokes-like problems is the indefiniteness of the algebraic system due to the saddle-point nature of the mixed discretization, and is clearly seen by applying Sylvester Law of inertia to the congruence transform of the Stokes discretization [29]. In order to solve a Stokes-like problem with iterative methods, two main approaches can be distinguished: whether the problem is solved as a whole, or “coupled”, or it is solved in a segregated way, for the pressure and then for the velocity [12]. In the first category, two possibilities for a Krylov-based iterative solver which take into account the symmetry of the problem are the MINRES and SYMMLQ solvers [47]. In the second category, the Uzawa method [2] allows to replace a minimisation constrained problem with a sequence of minimization problems without constraints. However, it appears that in both approaches, the bulk of the computation is concentrated on a good approximation of the PSC, which mimics the Uzawa operator at the discrete level. Applying block Gaussian elimination to (11), the PSC for the generalized Stokes problem reads:

$$\mathbf{B}\mathbf{A}^{-1}\mathbf{B}^T \mathbf{P} = \mathbf{B}\mathbf{A}^{-1}\tilde{\mathbf{F}}_V. \quad (14)$$

Matrix  $\mathbf{B}\mathbf{A}^{-1}\mathbf{B}^T$  is SPD up to the hydrostatic pressure mode. A conjugate gradient applied to (14) is the basis of the preconditioned Conjugate gradient Uzawa (PCGU) preconised in [13, 19, 65] and references therein. Originally, the Uzawa method is a stationary Richardson method at the continuous level [56]. The Uzawa operator reads [25]:

$$U_z(p) = -\nabla \cdot \left( \frac{\rho_0}{\delta t} \mathbf{I}_V - \theta \mu \Delta_V \right)^{-1} \nabla p, \quad (15)$$

$$= -\nabla \cdot \left( \frac{\rho_0}{\delta t} \mathbf{I}_V - \theta \mu \Delta_V \right)^{-1} \mathbf{f}_V, \quad (16)$$

where  $\mathbf{I}_V$  is the identity operator and  $\Delta_V$  is the Laplace operator in the velocity space. This is the equivalent of (14) at the continuous level.

The idea of [19] consists in distinguishing two separate behaviours of the Uzawa operator, whether the mesh Reynolds number defined as:

$$\text{Re}_m = \frac{\rho_0 h_K^2}{\mu \theta \delta t}, \quad (17)$$

is high or not, where  $h_K$  is an element reference length. In the first case, this gives:

$$\mathbf{A} \approx \frac{\rho_0}{\delta t} \mathbf{I}_V, \quad (18)$$

as the inertial part is preponderant over the diffusive part, so that:

$$U_z(p) \approx -\nabla \cdot \left( \frac{\rho_0}{\delta t} \mathbf{I}_V \right)^{-1} \nabla p \approx -\frac{\delta t}{\rho_0} \Delta p. \quad (19)$$

And in the second case:

$$\mathbf{A} \approx -\theta \mu \Delta_V, \quad (20)$$

so that:

$$U_z(p) \approx -\nabla \cdot (-\theta \mu \Delta_V)^{-1} \nabla p \approx \frac{1}{\theta \mu} p. \quad (21)$$

Finally, the proposed preconditioner of the Uzawa operator at the continuous level is:

$$\mathbf{C}^{-1} = \theta \mu \mathbf{M}_p^{-1} + \frac{\rho_0}{\delta t} \mathbf{K}_p^{-1}, \quad (22)$$

At the discrete level, the preconditioner writes:

$$\mathbf{C}^{-1} = \theta \mu \mathbf{M}_p^{-1} + \frac{\rho_0}{\delta t} \mathbf{K}_p^{-1}, \quad (23)$$

where  $\mathbf{M}_p$  and  $\mathbf{K}_p$  are the mass and Laplacian matrix in the pressure space. This idea relies on a Fourier analysis to simulate the multigrid efficiency at all frequencies. Later, [17] and [43] proved the uniform boundedness of the condition number of the preconditioned system with respect to the spatial discretization  $h$  and the time step  $\delta t$ . For the classical Stokes problem, the use of the inf-sup condition and the boundedness of the operator  $\mathbf{B}$  give the well-known bounds:

$$\exists c_1, c_2 > 0, c_1 \leq \frac{(\mathbf{B}\mathbf{A}^{-1}\mathbf{B}^T q, q)}{\left(\frac{1}{\mu} \mathbf{M}_p q, q\right)} \leq c_2 \quad \forall q \in \mathcal{R}^{n_p}, \quad (24)$$

proving the nice behaviour of the Uzawa operator independently of the mesh size  $h$ , and the spectral equivalence between the Uzawa operator and the finite element pressure mass matrix. For the non stationary case, the inf-sup with the  $\mathbf{H}_0^1$  norm for the velocity and  $L_0^2$  norm for the pressure still holds but do not reflect the behaviour of the Uzawa operator. The idea of [43] is to use a norm depending on the temporal

discretization parameter, the other parameters such as density and viscosity being included in this parameter, to extend equation (24). Proving the inf-sup condition and the boundedness of operator  $\mathbf{B}$  in this particular norm extends equation (24) to the non stationary case and gives:

$$\exists c_3, c_4 > 0, c_3 \leq \frac{(\mathbf{B}\mathbf{A}^{-1}\mathbf{B}^T q, q)}{\left(\left(\theta \mu \mathbf{M}_p^{-1} + \frac{\rho_0}{\delta t} \mathbf{K}_p^{-1}\right)^{-1} q, q\right)} \leq c_4 \quad \forall q \in \mathcal{R}^{n_p}, \quad (25)$$

which reflects the spectral equivalence between the above mentioned preconditioner and the Uzawa operator or, what is the same, the independence of the Uzawa operator respect to the mesh size and the physical and temporal parameters, when used with the preconditioner.

A variant of the last preconditioner, already described in [19], is the use of the so-called compatible or discrete Laplacian [54], which reads:

$$\tilde{\mathbf{C}}^{-1} = \theta \mu \mathbf{M}_p^{-1} + \frac{\rho_0}{\delta t} (\mathbf{B}\mathbf{M}_V^{-1}\mathbf{B}^T)^{-1}. \quad (26)$$

An appealing advantage of this discretization is that, as it will be seen later, it opens the way to a FS with discontinuous pressure elements. As a direct extension, the PCGU can now be applied to discontinuous pressure elements, as in [13]. The use of the compatible Laplacian reflects the discretization of the Laplacian by a mixed formulation:

$$\begin{pmatrix} \mathbf{M}_V & \mathbf{B}^T \\ \mathbf{B} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{V} \\ \mathbf{P} \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ -\mathbf{F}_p \end{pmatrix}, \quad (27)$$

where the natural boundary condition is  $p = 0$  on the Neumann boundary of the velocity, due to the integration by parts performed in the mixed formulation (27). So that, as noted in [65], with Neuman boundaries for the velocities, matrix  $\mathbf{B}\mathbf{M}_V^{-1}\mathbf{B}^T$  is SPD as is matrix  $\mathbf{B}\mathbf{A}^{-1}\mathbf{B}^T$  with the same boundary conditions, and not semidefinite positive as is the standard Laplacian. On Dirichlet boundaries for the velocity, matrix  $\mathbf{B}\mathbf{M}_V^{-1}\mathbf{B}^T$  does not provide any wrong boundary condition. To sum up, the most important property is that the pressure does not need a Dirichlet condition on the free surface as it must be with the standard Laplacian discretization. The superiority of this approach will be proved in the last section, but was already noticed in [19], particularly when the mesh Reynolds number tends to zero. It must be emphasized that the examples provided in [17, 19, 43] do not take into account Neumann boundary conditions for the velocity. In case of free surfaces, Dirichlet conditions must be applied on the standard pressure Laplacian so that if only this preconditioner is used, the PCGU scheme cannot converge as the imposed pressure degrees of freedom can not reach the mass conservation, which is the residual of the PCGU. With the addition of the pressure mass matrix, the scheme converges, but much slowly than with the compatible Laplacian. However, the bandwidth is roughly three times the bandwidth of the standard Laplacian in 2D, and five times in 3D on a structured grid, as it involves the neighbours of the neighbours of each nodes in case of continuous pressures and the



neighbouring elements for each element in case of discontinuous pressure. A good preconditioner is then mandatory. Furthermore, it could be possible to take advantage of the symmetric form of the matrix to require only the action of  $\mathbf{B}$  or  $\mathbf{B}^T$  which is already needed for the Stokes solver, as will be shown later. Evidently, the mass matrix must be lumped to have an efficient discretization.

#### 4.2 Algebraic splitting and FS methods

As noted in [17], the PCGU requires the action of  $A^{-1}$  whereas the global approach only requires a spectrally equivalent preconditioner. However, both methods will need to iterate and, in particular, to apply the action of the above preconditioner to each iteration. On the other hand, algebraic splitting methods offer a good compromise between accuracy and efficiency [57]. In this work, FS methods are encompassed in the more general algebraic splitting schemes, even if discretization in time is performed before spatial discretization in the FS methods, as opposed to algebraic splitting, which rely on the algebraic monolithic system. This point of view is motivated by only considering the final algebraic system of equations. By FS, we mean that a standard Laplacian is used for the pressure Poisson equation, and no final update of the pressure is performed. In [65], the equivalence between the continuous FS method and one iteration of the PSC preconditioned by the standard Laplacian was already noticed. Due to the previous discussion on the behaviour of the Uzawa operator and on the imposition of the pressure degrees on the free surface, it seems natural to replace the standard Laplacian by the compatible one and to use a pressure mass matrix to approximate the Uzawa operator at low mesh Reynolds number.

A very interesting scheme already proposed in [65] balances the accuracy of the PSC and the efficiency of a second order algebraic splitting. One solve reads:

$$\left(\frac{\rho_0}{\delta t} \mathbf{M}_V + \theta \mu \mathbf{K}_V\right) \tilde{\mathbf{v}} = \mathbf{F}_V - \mathbf{B}^T \mathbf{P}^n + \left(\frac{\rho_0}{\delta t} \mathbf{M}_V - (1-\theta) \mu \mathbf{K}_V\right) \mathbf{v}^n, \quad (28)$$

$$\frac{\delta t}{\rho_0} \mathbf{B} \mathbf{M}_V^{-1} \mathbf{B}^T \Delta \mathbf{P} = \mathbf{B} \tilde{\mathbf{v}}, \quad (29)$$

$$\mathbf{v}^{n+1} = \tilde{\mathbf{v}} + \frac{\delta t}{\rho_0} \mathbf{M}_V^{-1} \mathbf{B}^T \Delta \mathbf{P}, \quad (30)$$

$$\mathbf{P}^{n+1} = \mathbf{P}^n + \Delta \mathbf{P} + \theta \mu \mathbf{M}_p^{-1} \mathbf{B} \tilde{\mathbf{v}}, \quad (31)$$

where  $\Delta P$  is the pressure increment between time  $t^n$  and  $t^{n+1}$ . Compared to a classical FS, the standard Laplacian has been replaced by the compatible Laplacian and the pressure has been updated to take into account the behaviour of the PSC in case of low mesh Reynolds number. It mainly consists of one iteration of the PCGU scheme, as noted in [65], except that Eq. (30) has been added to perform the projection step, which gives an exact discretely divergence free velocity due to the compatible Laplacian. However, to go just one step further, the PCG algorithm is recalled on Fig. 1. The

Given  $x_0$ :

$$r_0 = b - Ax_0 \quad (1a)$$

$$z_0 = M^{-1}r_0 \quad (1b)$$

$$p_0 = z_0 \quad (1c)$$

While(not converged):

$$\alpha_j = (r_j, z_j) / (Ap_j, p_j) \quad (1d)$$

$$x_{j+1} = x_j + \alpha_j p_j \quad (1e)$$

$$r_{j+1} = r_j - \alpha_j Ap_j \quad (1f)$$

$$z_{j+1} = M^{-1}r_{j+1} \quad (1g)$$

$$\beta_j = (r_{j+1}, z_{j+1}) / (r_j, z_j) \quad (1h)$$

$$p_{j+1} = z_{j+1} + \beta_j p_j \quad (1i)$$

Fig. 1 The preconditioned conjugate gradient algorithm

implementation is the one proposed in [56]. As usual,  $x_0$  is the initial guess,  $r_j$  the residual,  $z_j$  the residual of the preconditioned system and  $p_j$  the search direction at iteration  $j$ . Applying the PCG on the PSC Eq. (14) and rearranging the terms with the previously mentioned preconditioner (see [4] for details), one sees that the algebraic splitting corresponds exactly to the initialisation process Eqs. (32a)–(32c) only. The final pressure of the MAS corresponds to the initial preconditioned residual, or equivalently to the first search direction. Furthermore, in the PCG algorithm, the first update of the unknown happens only at Eq. (32c). So that, in order for the first approximation to coincide with the final pressure value of the MAS, one must have  $\alpha_0 = 1$ . As  $\alpha_j$  represents the value which maximises the value of the quadratic functional associated with  $A$  in the  $p_j$  direction, or what is the same, brings the residuals orthogonal between each other, the pressure value obtained by the MAS is not in general the optimal value obtained from the previous steps if  $M \neq I$  or  $\theta \neq 1$ . A different algebraic splitting could then be constructed by computing the value of  $\alpha$  to update the pressure. It requires however one more matrix multiplication, or in the PSC context, one more inversion of the velocity system, which can not be used if one wants to verify the mass conservation. This would be very near to the Yosida method proposed in [54] with the complete preconditioner, as the final velocity would be the same and a pressure update would be performed.

More important are the controversial boundary conditions on pressure. As the FS does not reach the actualization step of the PCGU, the final pressure-like result of the FS is not an approximation of the pressure but the residual of the preconditioned system, so that boundary conditions for the pressure are the one inherited from the preconditioners. Whereas boundary conditions can be rather freely chosen for the preconditioner, they produce the serious well-known drawback in the classical FS. A remedy is to change the preconditioner used, as proposed in this section, and also considered in [53, 64], but the main reason is that the update of the pressure in the PCG is not performed. As will be seen in the last section, in case of a classical FS, it is better to impose the pressure on the free surface to the values given by the right-hand side of (29).

A straightforward extension of the PCGU and the algebraic splitting is the scheme used in this work, where the PCGU is used as a basic solver. If the convergence rate is too slow, or the maximum number of iteration is reached, the velocity iterate goes to the projection step (30), and the pressure is ultimately updated by Eq. (31). A complementary possibility is to evaluate the Reynolds mesh number and then activate or deactivate the proposed preconditioners. Concerning the velocity solve, a Symmetric Gauss Seidel (SGS) preconditioner implemented with Eisenstat's trick [28] is used to solve the velocity system.

*Remark 3* To follow the equivalence between bubble and stabilization methods of Remark 2, it is interesting to condensate the bubble element in the monolithic scheme [4], and rewrite the monolithic scheme as a second order FS as it has just been discussed, to appreciate how the bubble stabilises a second order FS, known to be unstable [22].

#### 4.3 A preconditioner for the preconditioner

As noted in [42], the costly part of a classical FS at high mesh Reynolds number is the Laplacian solve, mainly because if a small time step is used, the velocity system is very well conditioned whereas the time step has no influence on the pressure Laplacian. The same happens for the compatible Laplacian.

As already mentioned, the use of the compatible Laplacian  $\mathbf{B}\mathbf{M}_V^{-1}\mathbf{B}^T\mathbf{P} = \mathbf{F}_p$  reflects the discretization of the Laplacian by a mixed formulation. As noted in [44], the function spaces of the generalized Stokes problem and the mixed Laplacian are different, so that a stable Stokes element could be unstable for mixed Laplacian. The condition number of the condition number of system (27) relies on the weak inf-sup condition:

$$\delta \leq \min_{q \neq 1} \max_{v \neq 0} \frac{|(q, \nabla \cdot v)|}{\|v\| \|\nabla q\|} \quad (32)$$

and not on the “classical” inf-sup condition. This condition has been used in [66] to prove the classical inf-sup under adequate assumptions for the Taylor–Hood element. The matrix form of this condition reads:

$$\delta^2 = \min_{q \neq 1} \frac{(\mathbf{B}\mathbf{M}_V^{-1}\mathbf{B}^T q, q)}{(\mathbf{K}_p q, q)}. \quad (33)$$

In [43], it is shown that, in particular, this condition is met for the bubble element and for the  $P2/P0$  by generalizing the norm of the pressure for discontinuous approximations. The same arguments can be used to prove (32) for the  $P1+ / P1$ . Equation (32) gives the invertibility of matrix  $\mathbf{B}\mathbf{M}_V^{-1}\mathbf{B}^T$  and, by using the boundedness of operator  $\mathbf{B}$  with these “wrong” norms, proves that matrix  $\mathbf{B}\mathbf{M}_V^{-1}\mathbf{B}^T$  is spectrally equivalent to a Laplacian so that, in both cases of standard or compatible discretization, the condition number behaves like  $O(h^{-2})$  [6]. It is SPD up to a hydrostatic pressure mode, so that it is classically solved by a preconditioned conjugate gradient (PCG).

If the matrix must be assembled, it could be created globally as a sparse matrix product, or advantage could be taken of the grid so that the matrix is constructed locally by moving from pressure nodes to velocity nodes to pressure nodes in case of continuous pressure, or from element to velocity node to element in case of discontinuous pressure. In [19], the preconditioner is solved using a direct solver, which appears to be prohibitive for large meshes. In [13], no comment is made on how to solve the preconditioner. In [35, 54], it seems to be assembled at once, and solved by the BiCGStab. Finally in [57], a **QR** decomposition of  $\mathbf{B}\mathbf{M}_V^{-1/2}$  is advocated, as the compatible Laplacian pressure is solved twice. The **QR** decomposition allows to perform two triangular solves per equation. Except in [19] for 2D examples, the inverse of the velocity mass matrix has been previously lumped in all other papers. A discussion on the error committed by the mass lumping in this case can be found in [35].

In this work, a different approach is adopted. Three main requirements could be formulated:

- only the action of matrix  $\mathbf{B}\mathbf{M}_V^{-1}\mathbf{B}^T$  must be needed to avoid to assembly and store this matrix, particularly in case of a Lagrangian description where all matrices must be assembled every non-linear iteration so that all ILU(k)-type preconditioners are discarded.
- the preconditioner must be robust and not fear a null pivot during factorization.
- evidently the preconditioner must be efficient in reducing the condition number of the system to be solved.

Mainly, solving (29) amounts to solve a least square problem. In this work, the preconditioner chosen is the one presented in [10] and used for least-square problems in [11]. The preconditioner is based on an incomplete  $\mathbf{B}\mathbf{M}_V^{-1}\mathbf{B}^T$ -orthogonalization by a modified Gram–Schmidt [56] procedure, and only requires the action of the matrix and not its factorization. Furthermore, due to the special pivot used during the  $\mathbf{B}\mathbf{M}_V^{-1}\mathbf{B}^T$ -orthogonalization, it can not become null. It is proved in [10] that the Cholesky factors can be recovered from the orthogonalization process. In order to conserve a sparse preconditioner, a dropping rule is applied on the  $\mathbf{B}\mathbf{M}_V^{-1}\mathbf{B}^T$ -orthogonalized vectors. A second threshold is applied on the value of the  $\mathbf{L}\mathbf{L}^T$  factor in order to have a sparser triangular solve. As it will be seen, this second threshold is particularly efficient to conserve the relevant factors, keeping the preconditioner very sparse, resulting in a low storage and very efficient preconditioner. In the extreme case of no fill-in allowed, the SAINV preconditioner, the predecessor of the RIF preconditioner, is equivalent to a Jacobi preconditioner, and the RIF preconditioner to a SGS one. The version used in this work is the right-looking variant or rank-one update. It must be noticed that, in order for the preconditioner to be efficient, matrix-vector products must be done in *sparse–sparse* mode [56]. Storing matrix  $\mathbf{B}$  gives automatically the possibility to perform these products with matrix  $\mathbf{B}^T$ . Numerical results to illustrate the efficiency of the preconditioner are provided in the last section. The main reason of its success is that the dropping is applied on the inverse of the incomplete factors

instead of the factors themselves, so that  $\|\mathbf{I} - \mathbf{L}^{-1}\mathbf{A}\mathbf{L}^{-T}\|$  is minimized instead of  $\|\mathbf{A} - \mathbf{L}\mathbf{L}^T\|$ . The triangular  $\mathbf{L}\mathbf{L}^T$  solve is then much more stable (see e.g. [16] and references therein). All three requirements are then fully met. In order to have a least-square problem, Eq. (29) is replaced by:

$$\frac{\delta t}{\rho_0} \tilde{\mathbf{B}}\tilde{\mathbf{B}}^T \Delta \mathbf{P} = \mathbf{B}\tilde{\mathbf{V}}, \quad (34)$$

where:

$$\tilde{\mathbf{B}} = \mathbf{B}\mathbf{M}_L^{-1/2}, \quad (35)$$

with  $\mathbf{M}_L$  the lumped diagonal velocity mass matrix, as  $\mathbf{M}_L > 0$  for the  $P1 + /P1$  and the  $P1 + +/P1$  elements. Furthermore, a scaling is applied on the matrix to have an almost independent threshold value so that the system to be solved is:

$$\frac{\delta t}{\rho_0} \tilde{\mathbf{B}}\tilde{\mathbf{B}}^T \Delta \tilde{\mathbf{P}} = \mathbf{D}^{-1/2} \mathbf{B}\tilde{\mathbf{V}}, \quad (36)$$

where:

$$\tilde{\mathbf{B}} = \mathbf{D}^{-1/2} \mathbf{B}\mathbf{M}_L^{-1/2}, \quad (37)$$

with  $\mathbf{D}$  the diagonal of  $\mathbf{B}\mathbf{M}_L^{-1}\mathbf{B}^T$  and  $\tilde{\mathbf{P}} = \mathbf{D}^{1/2}\mathbf{P}$ .

#### 4.4 Mass lumping

In this section, a special comment on mass lumping is emphasized. The first step of the FS, or of the algebraic splitting, consists in solving a discrete elliptic or parabolic equation with a mass and a stiffness matrix. It is well-known that such discretization could not verify a discrete maximum principle under the appropriate assumptions [21]. To the authors' knowledge, this is the only known problem reported with the FS or the algebraic splitting in the literature, perhaps because of the fact that a high mesh Reynolds number could be avoided in the Eulerian formulation by a larger time step.

However, solving a classical dam break problem with water with a density of  $10^3 \text{ kg m}^{-3}$ , a characteristic mesh length of  $10^{-1} \text{ m}$ , a time step of  $10^{-3} \text{ s}$  and a dynamic viscosity of  $10^{-3} \text{ kg m}^{-1} \text{ s}^{-1}$  implies a mesh Reynolds number of  $10^7$  so that the velocity Laplacian contribution to matrix  $\mathbf{A}$  in Eq. (12) is almost inexistent. The velocity system amounts then to solve a mass matrix scaled by the density and the inverse of the time step. As noted in [21, 62], the finite element mass matrix for linear elements does not possess a maximum principle property as it is not a  $M$ -matrix. At the opposite, the sum of the lumped mass matrix with the velocity Laplacian, scaled by the viscosity, possesses this property assuming that no angle in the triangulation is bigger than  $\pi/2$ . For such high values of the mesh Reynolds number, instabilities were observed and the replacement by the lumped mass matrix amounts to be an efficient and accurate solution, as well as allowing the iterative solver to converge faster. Here, the discretization under consideration is not exactly performed by linear elements as the bubble shape function is added, which can produce non negative extra diagonal terms.

Indeed, it is equivalent to solve the same problem with linear elements on a grid where each element has been decomposed in three triangles in two dimensions and four tetrahedras in 3D, so that at the bubble nodes, the angles are greater than  $\pi/2$ . The mass matrix is still not a  $M$ -matrix and the stiffness matrix could have lost this property.

Alternatively, the introduction of the bubble in a reaction-diffusion problem has been interpreted in [30] as a stabilization applied to the reactive term, albeit not sufficient to eliminate the wiggles. From a different point of view, the stabilization comes from the fact that, as the mesh is finer, the reactive term is less strong than with the original mesh. Numerical experiments agree with the results of [30] so that lumping the mass matrix at high mesh Reynolds number is then still a necessity. At low mesh Reynolds number, numerical experiments seem to indicate that the maximum principle is still verified with the stiffness matrix without lumping the mass matrix.

This phenomenon was not observed for the  $P1 + +/P1$  mass matrix, which contains negative extra diagonal terms. In the same way, replacing the “hat function” by the cubic bubble implies to use a 13 Gauss points formula to integrate exactly the 6th degree mass matrix. Using reduced integration of 6 points reduces the oscillations, and using a three points formula eliminates the wiggles. Lumping the mass matrix is known to deteriorate the solution in some cases [59], but as mentioned in [37] the matching of the second order Crank-Nicolson time integration and the mass lumping constitutes an appropriate choice. Therefore, the classical drawback of the Uzawa scheme which needs an accurate inversion of matrix  $\mathbf{A}$  is somehow limited to low mesh Reynolds number, as much as the velocity mass matrix is lumped, which accelerates further the convergence of the PCG for the velocity system.

#### 4.5 Mass conservation

This section is the *raison d'être* of the solver part, and constitutes the main improvement with respect to previous work [5]. Mass conservation is a difficult problem, especially during a large numerical simulation. The non respect of it happens equally in an Eulerian or a Lagrangian framework. However, as the domain changes in a Lagrangian formulation, it is more easily perceptible.

Discontinuous pressure elements have the attractive property to verify locally the conservation equation. It means that the weak divergence will be equal to zero to the machine precision on each element. However, as already noted in remark (1), writing the equation of mass conservation in a traditional Eulerian way implies that a time derivation has already been performed, so that only an accurate time integration will imply that a small value of the weak divergence will produce a Jacobian value of 1 for each element in case of discontinuous pressure or on the patch of each node in case of continuous pressure. A reliable measure of the compliance of the mass conservation has been proposed in [48] by computing:



$$\max_K \left| \int_K \operatorname{div}_x \mathbf{v}_K dK \right|, \quad (38)$$

where  $\mathbf{V}_K$  is the velocity vector on element  $K$ . This measure is obtained at no additional cost for discontinuous pressure elements by using the PCGU, as convergence is measured by the norm of the residual  $\mathbf{B}\mathbf{V}$  which contains the precedent measure for the constant part of the pressure. For continuous pressure elements, apart from the norm of the residual, we propose the measure:

$$\sum_i \int_{\text{Patch}} \Psi_i \operatorname{div}_x \mathbf{v}_j d\Omega_i = \int_{\Omega} \operatorname{div}_x \mathbf{v} d\Omega, \quad (39)$$

where the integration domain is the patch of node  $i$ , indeed all the elements which contain  $i$ , and  $\mathbf{v}_j$  is the velocity vector of all the neighbouring nodes plus  $i$  of node  $i$ . It is not a norm, but represents the global mass conservation, which is the best that can be achieved by continuous pressure elements. Results with both the Euclidian norm of the residual and the last measure are given in the last section.

By using the classical FS, the introduction of the Laplacian pressure matrix produces the controversial Neuman boundary condition for the normal pressure derivative, but above all in case of free surface problems, implies to impose the pressure at the free surface as a Dirichlet condition. Taking the weak divergence of the end-of-step velocity, the velocity degrees of freedom belonging to the free surface can not verify the null divergence as the pressure nodes associated with the mass conservation equation have been removed from the Laplacian equation. Furthermore, during a dam break, elements or groups of elements can fly away and mix thereafter. Imposing a null pressure on all the nodes of one element produces a null pressure in the whole element and the impossibility to satisfy the mass conservation, as will be seen below.

## 5 The thermal problem

In this section, a classical discretization of the temperature is presented. One common numerical problem in mould filling appears with the thermal shock between the cold mould and the boiling casting, so that wiggles constantly appear with a classical discretization. A short review of possible remedies is commented in [49]. A mixed discretization temperature/heat flux may avoid this problem. In order to compare with the last section, it must be noted that, contrarily to the generalized Stokes system (11), an inf-sup condition is not required due to the presence of the SPD temperature mass matrix [33, 62], so that an equal order discretization appears to be very attractive computationally. Furthermore, applying a block-Gaussian elimination as in the previous part, the same kind of system must be solved and the solver of the previous section is readily available. For example, the mixed temperature equation to be solved reads:

$$\left( \rho_0 C \frac{\mathbf{M}_T}{\delta t} + \theta \kappa \mathbf{B} \mathbf{M}_q^{-1} \mathbf{B}^T \right) \mathbf{T}^{n+1} = \tilde{\mathbf{F}}_T. \quad (40)$$

Matrix  $\left( \frac{\rho_0 C}{\delta t} \mathbf{M}_T + \theta \kappa \mathbf{B} \mathbf{M}_q^{-1} \mathbf{B}^T \right)$  is SPD so that it can be solved by PCG. However, compared to Eq. (29), the presence of the temperature mass matrix provides a much better conditioned operator so that, after having lumped the flux mass matrix, it could be interesting to solve it as in the previous section. Nonetheless, as mentioned in Sect. 4.4, the temperature mass lumping does not only produce an additional diffusion balanced by the Crank–Nicolson time integrator but guarantees under reasonable mesh assumptions a monotonic scheme as the discretization is performed only with linear elements here. This valuable property compared with the increased storage of the mixed temperature/heat flux discretization makes the classical scheme much more attractive. Furthermore, at high mesh Reynolds number, the matrix becomes much more diagonally dominant and only a few iterations are necessary to converge with the PCG.

## 6 Numerical examples

In this section, the theoretical predictions of the spatial and time discretizations described in the previous sections for the generalized Stokes and thermal problem are illustrated on various numerical examples.

### 6.1 Preconditioner results

The efficiency of the algebraic splitting relies mainly on the efficiency of the RIF preconditioner. It must be noted that, for the PCGU, the preconditioner is assembled only once during the Uzawa iteration process, so that its cost is amortized over the iterations.

Its performances are illustrated in Tables 1 and 2 on a 3D dam break example presented below with a  $P1 + /P1$  element. Each result consists in the storage requirement of the RIF normalized with the one of the  $\tilde{\mathbf{B}}\tilde{\mathbf{B}}^T$  matrix in the upper left corner, the computational time for the RIF construction in seconds in the lower left corner, the iteration number of the PCG applied to matrix  $\tilde{\mathbf{B}}\tilde{\mathbf{B}}^T$  at the upper right corner, and the computational time in seconds of the PCG in the lower

**Table 1** RIF preconditioner with various pre and post dropping values

SAINV drop tolerance	Post filtration threshold			
	0		1e-3	
0.001	3.2	22	0.91	23
	207	3	207	2
0.01	1.2	25	0.8	25
	27	3	27	2
0.1	0.51	39	0.5	39
	1	3	1	3

The columns represent various SAINV drop tolerances and lines various post filtration thresholds. For each case, the storage requirement is reported in the upper left corner, the computational time in seconds for the RIF construction in the lower left, the iteration number of the PCG in the upper right, and the computational time in seconds for the PCG only in the lower right



**Table 2** RIF preconditioner

SAINV drop tolerance	Post filtration threshold					
	1e-2		1e-1		1	
0.001	0.41	29	0.04	73	0.02	101
	206	2	207	4	206	5
0.01	0.4	31	0.04	72	0.02	101
	27	3	26	5	27	7
0.1	0.36	42	0.04	73	0.02	102
	1	4	1	5	1	7

right corner. These results are reported with various thresholds applied on the inverse factors and then on the factors themselves as a post filtration for a tolerance of  $\|b\|/\|r^n\|$  of  $10^{-13}$  for the PCG.

During the factorization, for each  $\mathbf{Z}_i$  vector forming the  $\tilde{\mathbf{B}}\tilde{\mathbf{B}}^T$ -orthogonal basis, scalar products must be done with others  $\mathbf{Z}_j$  vectors. A computation for each vector against the others would give a complexity of the process of  $O(n^2)$ , making the whole process very inefficient. As the  $\tilde{\mathbf{B}}^T\mathbf{Z}_i$  vectors have only few components, advantage can be taken from the sparsity in order to compute for each  $\mathbf{Z}_i$  only the  $\mathbf{Z}_j$ 's that will give a non vanishing scalar product. This task is achieved by computing in a first pass the neighbours of the neighbours for each point in the continuous pressure case, and the neighbours of each element in the discontinuous pressure case.

As already mentioned, the  $\tilde{\mathbf{B}}\tilde{\mathbf{B}}^T$  matrix is spectrally equivalent to a Laplacian, which means that it is bad conditioned but not ill-conditioned. For example, the PCG converges without preconditioning. The results reported in Tables 1 and 2 illustrate the nice behaviour of the RIF preconditioner. The preconditioner behaves exactly as predicted; the bigger the tolerance pre or post, the better the iteration number. For the same example with the  $P1 + /P1$ , the RIF preconditioner reaches a gain in time of 30% with optimal parameter values with respect to the diagonal preconditioner. Finally, the low storage requirements due to the post-filtration makes RIF a very efficient preconditioner.

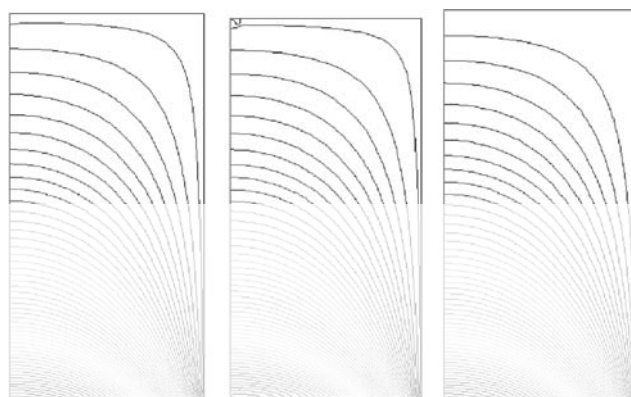
## 6.2 Behaviour of the Uzawa operator

The main goal of this section is to compare the results obtained by the classical FS, the MAS and a monolithic solution obtained with the fully converged PCGU on a dam break problem and on the no-flow problem.

### 6.2.1 2D dam break

The example studied is the first step at  $t = 0$  of a 2D dam break, which is a typical free surface problem. Initially, the fluid occupies a rectangle of dimensions  $0.3 \times 0.6$ , the tank is  $1 \times 1$  and the mesh is unstructured of size 0.01. The spatial discretization is performed with a  $P1 + /P1$  element. As explained in Sect. 4.1, the Laplacian preconditioner is expected to behave well with high mesh Reynolds number, the pressure mass matrix on low mesh Reynolds number, and the optimal

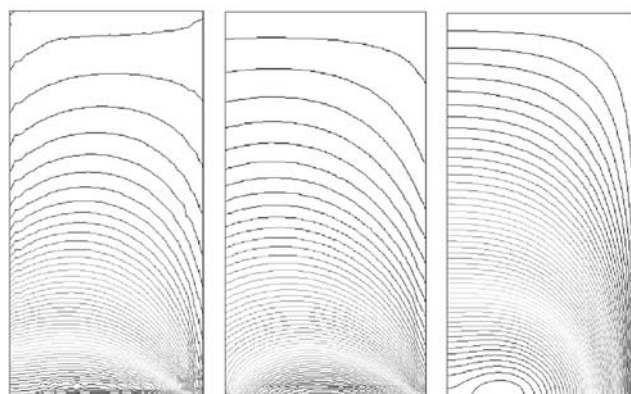
preconditioner on the whole range of mesh Reynolds number. Therefore, the comparison is performed at three different mesh Reynolds number regimes, a high, middle and low one. Results are displayed on Fig. 2. The first test is performed with  $Re_m = 10^5$ , the second with  $Re_m = 1$  and the third with  $Re_m = 0.01$ . For each row, from left to right the pictures represent the results of the PCGU, the MAS and the FS, where it



(a) PCGU, MAS and FS at  $Re_m = 10^5$



(b) PCGU, MAS and FS at  $Re_m = 1$



(c) PCGU, MAS and FS at  $Re_m = 10^{-2}$

**Fig. 2** Pressure isolines of a 2D dam break result for the PCGU, the modified algebraic splitting (MAS) and the fractional step (FS) at mesh Reynolds number  $\{10^5, 1, 10^{-2}\}$  at  $t = 0$  with a  $P1 + /P1$  element. The tank sides are at the left and at the bottom

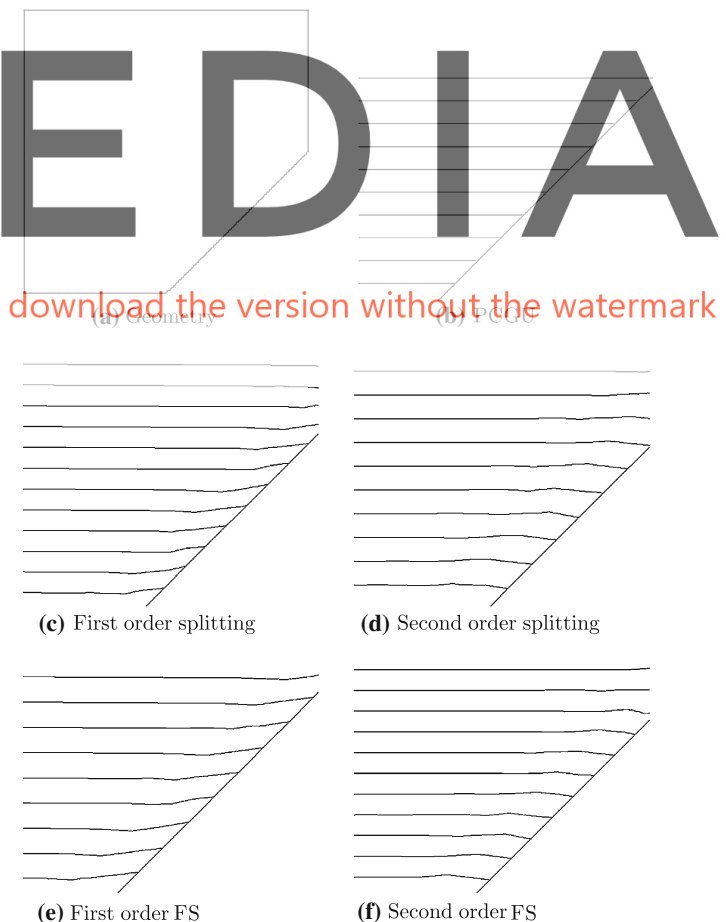
is reminded that a standard Laplacian is used for the pressure Poisson equation and no final update is performed for the pressure. As expected, at  $Re_m = 10^5$  the pressure isolines are in very good agreement for the MAS and the FS respect to the monolithic scheme, which explains the success of the FS. At  $Re_m = 1$ , slight differences begin to appear at the bottom of the wave between the MAS and the FS, the MAS being closer to the PCGU than the FS. At  $Re_m = 0.01$ , the results obtained by the Pressure Laplacian are still driven by the gravity force, so that they are very far from the PCGU results, making the solution inaccurate. The MAS stays rather close to the PCGU, providing a cheap approximation of the monolithic scheme.

**Remark 4** Here, emphasis is obviously put on free surface flows, due to the Lagrangian formulation. As mentioned in Sect. 4.2, the standard Laplacian preconditioning imposes the pressure as a Dirichlet boundary condition, and prohibits or delays the convergence of the PCGU. For this simple 2D example at  $Re_m = 1$ , the PCGU converges with a residual norm of  $10^{-13}$  with 17 iterations with the compatible Laplacian and the pressure mass matrix, whereas 91 iterations were necessary for the standard Laplacian and the pressure mass matrix. A better solution is to impose the standard Laplacian to the residual of the PCGU. The PCGU then converges with 29 iterations. However, advantage can not be taken for the FS because it equally fixes the pressure degrees of freedom at the free surface. It is possible that this procedure reduces the mass loss for standard FS. For the 3D example presented in the next section, the results are still more significant; at high  $Re_m$ , the PCGU with the compatible Laplacian needs only three iterations to converge with the pressure mass matrix, but 482 iterations of standard Laplacian are necessary by imposing the preconditioned system to zero at the free surface, and 390 iterations are obtained by imposing the preconditioned system to the residual value. Without Laplacian preconditioner, 95 iterations are necessary to reach convergence. With only the standard Laplacian, the norm of the residual does not decrease after 16 iterations, and the PCGU does not converge. These results show the difficulty of the standard Laplacian to maintain a null divergence on the free surface so that all the schemes with a standard Laplacian will produce an important mass loss around the free surface.

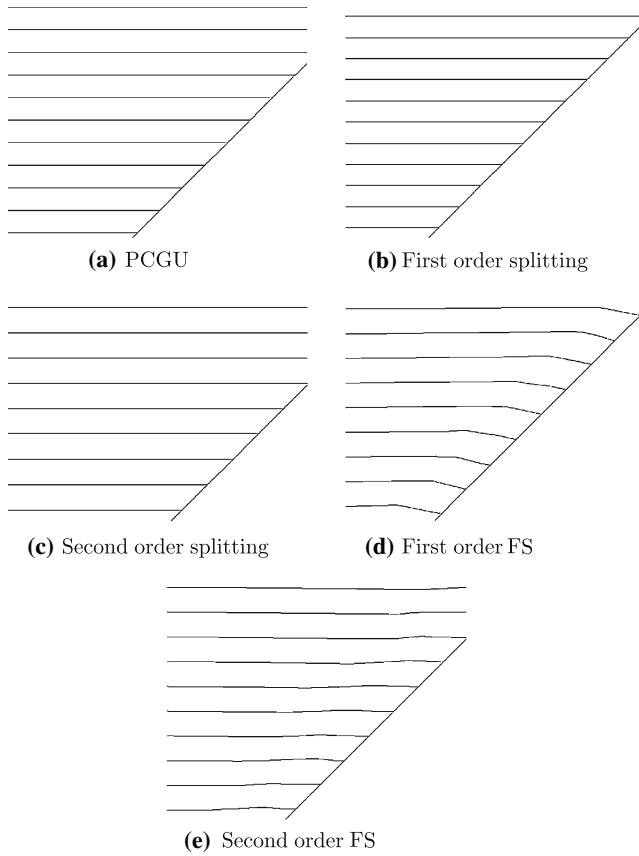
### 6.2.2 The no-flow test

Another interesting numerical example is the no-flow test [34], in order to verify the pressure boundary layer due to the FS, and to compare the end-of-step velocity of the FS and the algebraic splitting to the monolithic solution, where the exact solution is the hydrostatic pressure and a uniformly null velocity. Here, the fluid is water so that the mesh Reynolds number is high for a reasonable time step, namely  $Re_m = 10^5$  here. The geometry is such that one side is oblic, in order for the condition imposed on the normal derivative of the pressure to not coincide with the exact normal derivative on

the wall side. Results are performed with the PCGU, the algebraic splitting of first and second order, and the classical first and second order FS. Second order means that the pressure of the first order has been reinjected in the momentum equation. The tank dimension is  $2 \times 2$  and the oblic part corresponds to the diagonal of the  $1 \times 1$  lower right square. The spatial discretization has been performed with a  $P1 + / P1$  element, and the mesh is unstructured with size 0.05. Figure 3a shows the whole geometry whereas all other results are centred on the oblic side. In Fig. 3, a consistent velocity mass matrix is used in (12) whereas a lumped mass matrix is used in Fig. 4. The maximum velocities for the five examples with consistent and lumped mass matrix are depicted in Tables 3 and 4. As the solution belongs to the discrete spaces, the monolithic solution displays the exact solution without pressure boundary layer as expected. However, it is rather surprising to see the pressure isolines of the first order algebraic splitting bending around the oblic boundary with the consistent velocity mass matrix, nonetheless with a curvature which does not reflect the pressure boundary layer. Examining the maximum velocity norm for these examples, it is seen that instabilities



**Fig. 3** The no-flow test with consistent velocity mass matrix for the PCGU, first and second order splitting, and first and second order FS schemes. The pictures represent the geometry and a detail on the oblic side



**Fig. 4** The no-flow test with lumped velocity mass matrix for the PCGU, first and second order splitting, and first and second order FS schemes. The pictures represent a detail on the oblique side

**Table 3** Maximum velocity norm for the no-flow test with consistent and lumped mass matrix with  $Re_m = 10^5$  for the PCGU and the algebraic splitting

	PCGU	1st split	2nd split
Consistent	1e-14	5e-3	2e-3
Lumped	1e-13	1e-7	1e-12

**Table 4** Maximum velocity norm for the no-flow test with consistent and lumped mass matrix with  $Re_m = 10^5$  for the fractional step (FS)

	1st FS	2nd FS
Consistent	6e-3	1e-3
Lumped	8e-3	3e-3

are created by the consistent mass matrix which generate the bending of the pressure isolines for the algebraic splitting. In fact, the first step of the classical FS is a discretization of a parabolic equation, well-known for the instabilities as stated in Sect. 5. On the other hand, the lumped mass matrix gives a velocity guess without instabilities, as seen in Tables 3 and 4, and no pressure boundary layer due to a possible  $\frac{\partial p}{\partial n} = 0$  condition appear for the algebraic splitting whereas it is clearly present for the classical FS. Using a lower mesh Reynolds number stresses the bad preconditioning of the FS or the

algebraic splitting to the Uzawa operator, as explained for the last example. Whereas the solution is still the same, the maximum velocity norm is around 0.1 and does not produce the correct velocity field. The pressure is far better with the algebraic splitting and the scaled pressure mass matrix, but iterations to the monolithic scheme seem to be unavoidable.

*Remark 5* If the original velocity mass matrix is lumped, convergence of the PCGU preconditioned by the compatible Laplacian was observed in two iterations for a mesh Reynolds number of  $10^7$ . It is in perfect agreement with the theoretical behaviour as, for such a high value of mesh Reynolds number, the Uzawa operator contains almost only the inertial part. Inverting matrix  $BM_L^{-1}B^T$  is then almost equivalent to inverting the Uzawa operator. If one accepts to lump the velocity mass matrix, the solution of the monolithic scheme at high mesh Reynolds number is obtained very efficiently and constitutes an appealing scheme.

### 6.3 Mass conservation

The studied problem is the same as above, namely a dam break test, except that it is a 3D dam break and it is performed over 5 s of computational time. It involves breaking waves with water, strong contact and mixing so that it is a good test to compare mass loss. The tank geometry is  $10 \times 10 \times 10$  and initially the fluid volume is  $3 \times 3 \times 6$  at the corner. The mesh is unstructured with size 0.3. The initial Reynolds mesh number is  $10^5$ , which varies due to the variable time step chosen so that the Courant number be equal to 1. The results have been performed with three different elements:

- a  $P1/P1$  backward Euler stabilized by a first order FS.
- a  $P1++/P1$  with discontinuous linear pressure MAS.
- a  $P1++/P1$  with MAS, FS and PCGU.

The dam break results have not been displayed as other 3D dam break results are presented in the next section. In Table 5, the number of velocity and pressure degrees of freedom are reported. It must be noted that the bubble is condensed for the velocity system for the second element, and for the whole system for the third element. Values of the averaged minimal, maximum Jacobian for each element and the averaged Jacobian for all the elements are reported in Figs. 5, 6 and 7. The curves were smoothed using Bezier curves in sake of clarity. These values correspond to the values obtained for the convergence of the non linear process. As the nodes are convected with their displacement values, an isoparametric element is used in the non-linear iteration. The time step is variable, but of the same order for all the examples, around

**Table 5** Number of velocity and pressure nodes for the  $P1++/P1$ ,  $P1++/P1$  and  $P1/P1$  elements for a 3D dam break

Element	Velocity nodes	Pressure nodes
$P1++/P1$	163735	195940
$P1++/P1$	63345	14360
$P1/P1$	14360	14360

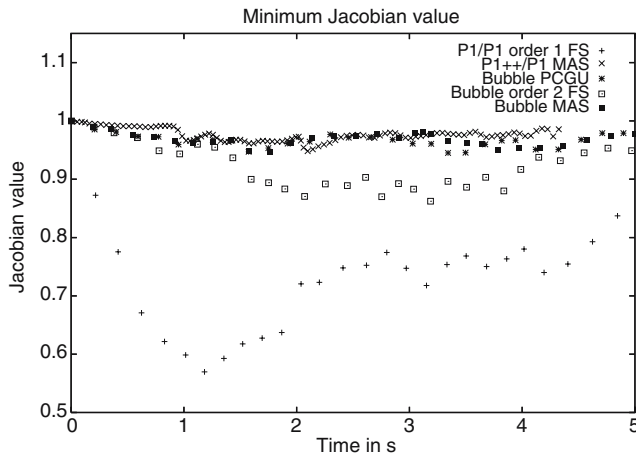


Fig. 5 Minimal Jacobian value for various elements and solvers

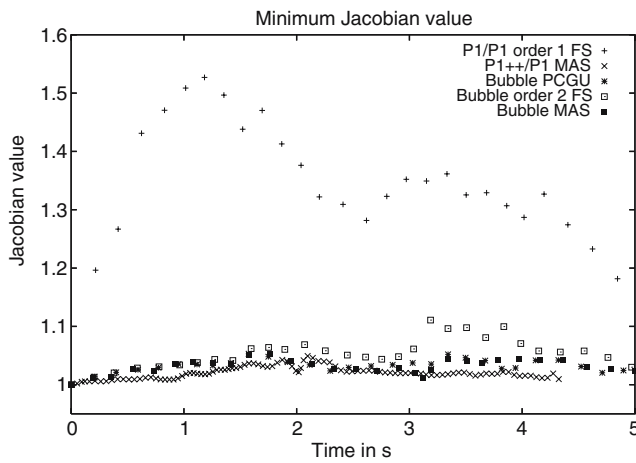


Fig. 6 Maximum Jacobian value for various elements and solvers

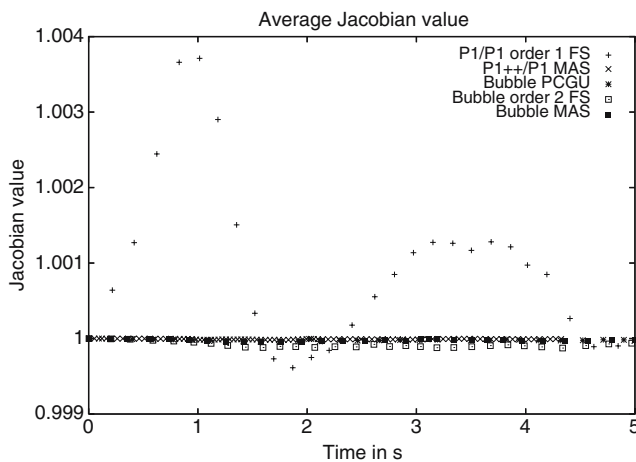


Fig. 7 Average Jacobian value for various elements and solvers

$10^{-3}$  s. For the  $P1++/P1$  element, the numerical integration is performed with 24 Gauss points, so that the terms involving the inf-sup condition are integrated with sufficient accuracy. It was observed that the velocity system could not

be solved with less Gauss points per elements. An isoparametric formulation is used for the  $P1++/P1$  so that the Jacobian takes into account all the shape functions, as the geometry of the bubble must also be interpolated. A negative Jacobian does not necessarily imply that the element screw itself but that the classical condition on the edges :

$$\|a_{i,K} - \tilde{a}_{i,\tilde{K}}\| \leq C h_K, \quad (41)$$

where  $a_{i,K}$  is the vertex of the element and  $\tilde{a}_{i,\tilde{K}}$  is the mid edge of the corresponding straight-sided element, is not fulfilled. It is remarkable that no problem was met during the computation, mainly explained by the small time steps used.

The computation for the linear elements and the bubble involved around two days of real time for a whole result of 20 s of computational time, whereas the  $P1++/P1$  was stopped after two weeks for less than 5 s of computational time. Due to the discontinuous pressures and the second order time integration, the Jacobian values are very near from 1 for the  $P1++/P1$  element in the three cases, as the discontinuous pressure implies a local mass conservation. However, as noted before, the elemental Jacobian value is not even weakly constrained to be 1 but rather the divergence of the velocity is weakly constrained to be null on each element. Here, time integration appears to be very important and the second order time integration of the  $P1++/P1$  seems to provide accurate Jacobian values very near from 1 up to  $10^{-5}$  for the averaged Jacobian value.

For the first order in time  $P1/P1$  element, results are completely different. Negative values of the Jacobian are even observed for the  $P1/P1$  element with the classical FS at longer time, which does not preclude the non linear convergence. This is not surprising, as the measured Jacobian values are always the one of the previous iteration, so that it has been checked that all the previous Jacobian were positive and that the non linear convergence has been reached, but not that the Jacobian of the last iteration were also all positive. The imposition of the pressure on the free surface coupled with a first order integration in time gives by far the worst overall result. It must be noted that over the 7,207 iterations performed in time, 348 backtrace iterations when the Jacobian of an element becomes negative were necessary to go back at the beginning of the time step whereas no backtrace iterations were performed for the PCGU and the MAS whatever the element.

The  $P1++/P1$  element gives good values of Jacobian also up to  $10^{-5}$  for the averaged Jacobian value, even if mass conservation is only applied on the patch of each node with the MAS and the monolithic. It can be seen that the results are slightly better with the PCGU, which was expected. However, the second order FS gives the worst results, better than the linear element due to the higher order time integration, but very far from the MAS and the PCGU.

The maximum value of the norm of the divergence and the proposed value to measure the mass loss is reported in Table 6 for the previous cases. The very high values of the FS are due to the cases where a group of elements have only free surfaces nodes, in which case the pressure is null on the



**Table 6** Maximum value of  $\|BV\|$  and the value  $\sum_i B_{ij} V_j$  for various elements and solvers

	$\ BV\ $	$\sum_i B_{ij} V_j$
P1/P1 1st order FS	$10^6$	$10^6$
P1++/P1 MAS	$4 \times 10^{-10}$	$2 \times 10^{-9}$
P1+/P1 PCGU	$1 \times 10^{-12}$	$5 \times 10^{-12}$
P1+/P1 2nd order FS	$10^6$	$10^6$
P1+/P1 MAS	$4 \times 10^{-14}$	$2 \times 10^{-12}$

whole domain, and no control could be achieved on the norm of the divergence on the whole domain. In the average, the value is around 1. For the PCGU case,  $\|BV\|$  is the residual of the PCGU so that it is verified up to the tolerance, which was fixed to  $10^{-12}$ . The global value is slightly less as it is not imposed by the algorithm, but really corresponds to the mass loss on the whole domain. For the MAS case, the values of  $\|BV\|$  are better as they correspond to the tolerance of the system involved with matrix  $\mathbf{BM}_V^{-1}\mathbf{B}^T$ , which was fixed to  $10^{-13}$ .

If until then, all the results coincide with the theoretical considerations, Fig. 8 gives some unexpected results. The volume loss has been reported as the difference of the volume of the straight elements before and after moving the mesh divided by the initial volume. It must be noticed that, as the alpha shape method [27] is used to find the fluid boundaries, it could be also responsible for a mass loss. However, the volume loss reported on Fig. 8 is *exclusively* measured as the difference between the volume of the straight elements between two time steps before remeshing. It is seen that, even if the P1++/P1 element provides a nice divergence norm, the computed volume loss is not as good as expected. It gives the best values but there is not an order of improvement, whereas it needs two orders more in time. It could be due to the fact that it does not take into account the curved surfaces of the element at the end-of step. The geometry becomes increasingly complicated if curved faces must be taken into account by the mesh generator at the next step so that only the volume loss of straight elements really matters. Further-

more, it is also seen that all the second order schemes do not give better volume conservation than the first order whereas the divergence and Jacobian measures were far better. As a  $\theta$ -scheme is used, the measured Jacobian corresponds to a domain considered between  $n$  and  $n+1$  whereas the volume is moved with the final computed displacement. This fact also explains the bad Jacobian values obtained by the first order scheme, as in this case, the Jacobian is computed with the whole displacement and not at an intermediate state where the displacement is less important. A second order backward differencing scheme could be a possible remedy but it deserves further investigation.

#### 6.4 3D dam break

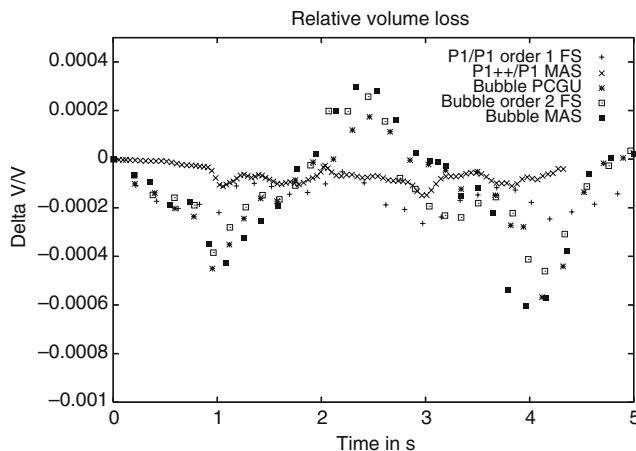
This example is the classical 3D dam break with a solid cube at the middle of the domain. The mesh is composed by  $2 \times 10^5$  nodes and around  $10^6$  elements with a size map concentrated around the solid cube. Thermal properties are the one of the water. Initially, the walls have a temperature of  $20^\circ\text{C}$  whereas the fluid has a temperature of  $60^\circ\text{C}$ . The temperature distribution is depicted at various time steps in Figs. 9 and 10. The example aims at illustrating the robustness of the method, and the accurate thermal convection due to the Lagrangian formulation.

## 7 Conclusion

Various numerical aspects of a fully non linear thermally semi-coupled incompressible Lagrangian formulation have been presented in this paper. First of all, the behaviour of the classical FS, the algebraic splitting and a variant with the introduction of the pressure mass matrix have been compared to conclude that the FS-like procedures are useful numerical tools if they are used in the appropriate mesh Reynolds number, as are the equivalent preconditioners for the Uzawa operator. The first or second order approximation is obviously asymptotic and care must be taken to the underlying constants, as they also depend on the mesh discretization, the dynamic viscosity and the density. However, the monolithic solution remains still a general and unavoidable issue. A new scheme has been presented to achieve the monolithic solution in case of high mesh Reynolds number with very few iterations if one accepts to lump the velocity mass matrix.

In the thermal field, the temperature mass lumping and the second order Crank–Nicolson time integration provide an accurate and very efficient solver. The semi-coupling has been illustrated on a rather complicated three dimensional dam break.

In order to compare mass conservation in a free surface context with Lagrangian description, the use of various spatial discretizations with continuous and discontinuous pressure has been implemented. However, it was observed that the introduction of discontinuous elements did not lead to much more accurate results, mainly because of the unavoidable use

**Fig. 8** Relative volume loss for various elements and solvers

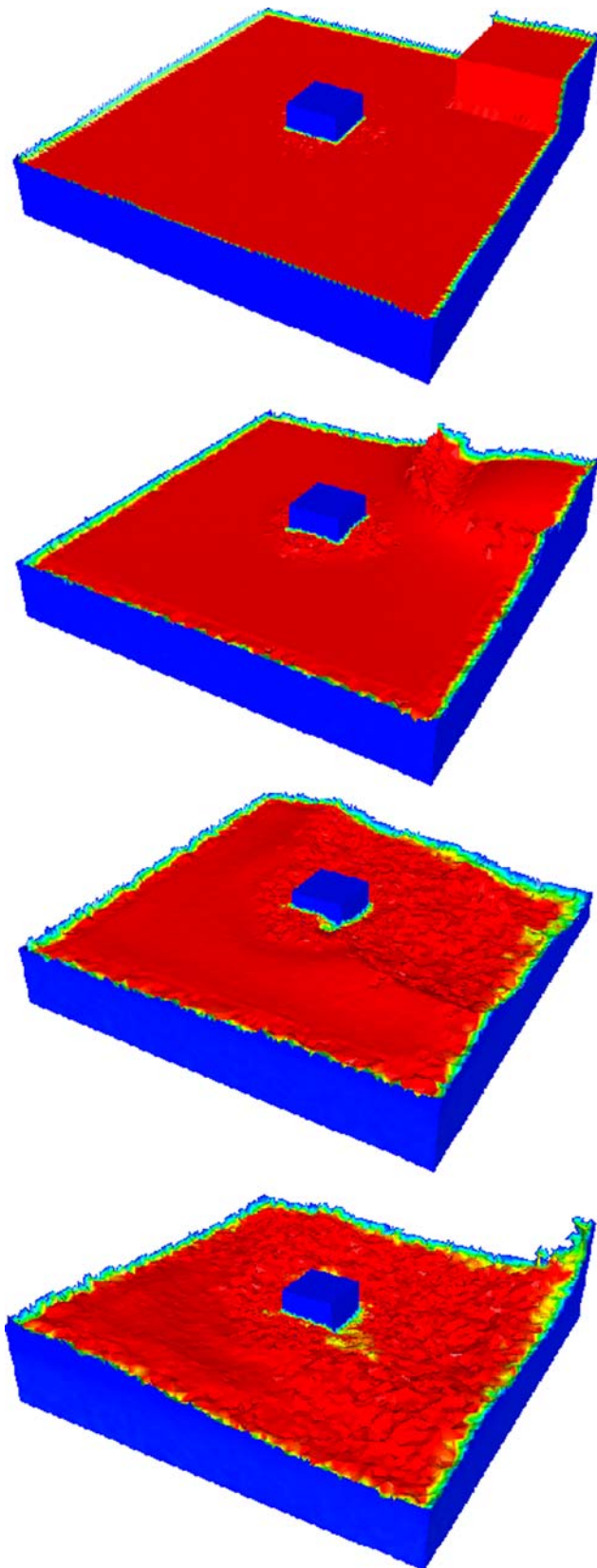


Fig. 9 3D dam break at  $t=\{0.001, 0.9, 2.3, 4\}$

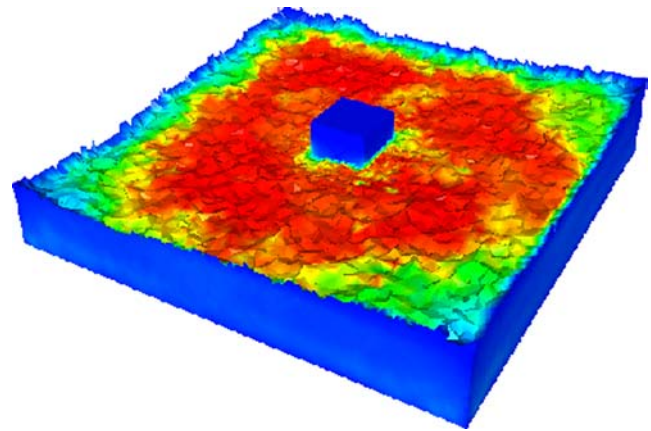


Fig. 10 3D dam break at  $t=40$  s

of isoparametric elements which generate volumes that can not be recovered by straight simplex elements at such a low tolerance, whereas the global volume of the latter measures what really matters. Furthermore, it implies an increase of two orders in time which, even if it provides better results for the same mesh, is easily outperformed by slightly increasing the degrees of freedom of  $C^0$  pressure elements. In view of these results, the bubble element still happens to constitute a correct solution, providing a very good compromise between accuracy and efficiency. The key point in mass conservation with free surface problems, at least from a divergence point of view, is that algebraic splitting methods leave boundary pressure conditions as a Neumann boundary condition. This result applies equally to an Eulerian description, but do not appear so clearly, except by computing the volume lost by moving the mesh with the end-of-step velocity. The “ultimate” mass conservation scheme must impose in its formulation that the Jacobian be equal to one, independently of the computational cost and the description, the time integration being as important as the norm of the divergence as the precision increases.

In view of the problems met in general, it is rather comfortable that no convective term appears in the discretization, which still more strengthens the numerical difficulties. Finally, various numerical examples have illustrated the robustness of the proposed approach. Under investigation is the application to these tools to the solidification problem and the full understanding of the rather poor volume conservation of the second order  $\theta$ -schemes tested in this work.

**Acknowledgements** Thanks are given to Michele Benzi for his always kind and very valuable answers during the implementation of the RIF preconditioner.

## References

1. Arnold DN, Brezzi F, Fortin M (1984) A stable finite element for the Stokes equations. *Calcolo* 21(4):337–344

2. Arrow K, Hurwicz L, Uzawa H (1958) Studies in linear and non-linear programming. Stanford University Press, Stanford
3. Ashby SF, Manteuffel TA, Saylor PE (1990) A taxonomy for conjugate gradient methods. *SIAM J Numer Anal* 27(6):1542–1568
4. Aubry R (2006) Incompressible Lagrangian fluid flow with thermal coupling. PhD thesis, Universidad Polit cnica de Catalunya
5. Aubry R, Idelsohn SR, O ate E (2005) Particle finite element method in fluid mechanics including thermal convection-diffusion. *Comput Struc* 83:1459–1475
6. Axelsson O, Barker VA (2001) Finite element solution of boundary value problems. Classics in applied mathematics. SIAM, Philadelphia
7. Baiocchi C, Brezzi F, Franca LP (1993) Virtual bubbles and the Galerkin-least-squares methods. *Comput Methods Appl Mech Eng* 105:125–141
8. Bank RE, Welfert BD (1990) A comparison between the mini element and the Petrov-Galerkin formulations for the generalized Stokes problem. *Comput Methods Appl Mech Eng* 83:61–68
9. Belytschko T, Liu WK, Moran B (2000) Nonlinear finite elements for continua and structures. Wiley, New York
10. Benzi M, Tuma M (2003) A robust incomplete factorization preconditioner for positive definite matrices. *Numer Linear Algebra Appl* 10:385–400
11. Benzi M, Tuma M (2003) A robust preconditioner with low memory requirements for large sparse least square problems. *Siam J Sci Comput* 25(2):499–512
12. Benzi M, Golub GH, Liesen J (2005) Numerical solution of saddle point problems. *Acta Numerica* 14:1–137
13. Bertrand F, Tanguy PA (2002) Krylov-based Uzawa algorithms for the solution of the Stokes equations using discontinuous-pressure tetrahedral finite elements. *J Comput Phys* 181(2):617–638
14. Bertrand F, Gadbois M, Tanguy PA (1992) Tetrahedral elements for fluid flow problems. *Int J Numer Methods Eng* 33:1251–1267
15. Blair Perot J (1993) An analysis of the fractional step method. *J Comput Phys* 108(1):51–58
16. Bollh fer M (2001) A robust ILU with pivoting based on monitoring the growth of the inverse factors. *Linear Algebra Appl* 338:201–218
17. Bramble JH, Pasciak JE (1997) Iterative techniques for time dependent Stokes problems. *Comput Math Appl* 33:13–30
18. Brezzi F, Fortin M (1991) Mixed and hybrid finite element methods. Springer, Berlin Heidelberg New York
19. Cahouet J, Chabard JP (1988) Some fast 3D finite element solvers for the generalized Stokes problem. *Int J Numer Methods Fluids* 8:869–895
20. Chorin AJ (1968) Numerical solution of the Navier–Stokes equations. *Math Comp* 22:745–762
21. Ciarlet PG, Raviart PA (1973) Maximum principle and uniform convergence for the finite element method. *Comput Methods Appl Mech Eng* 2:17–31
22. Codina R (2000) Pressure stability in fractional step finite element methods for incompressible flows. *J Comput Phys* 190:1579–1599
23. Crank J, Nicolson P (1947) A practical method for numerical evaluation of solutions of partial differential equations of the heat conduction type. *Proc Camb Phil Soc* 43:50–64
24. Crouzeix M, Raviart PA (1973) Conforming and nonconforming finite element methods for solving the stationary Stokes equations. *RAIRO Analyse Num rique*, pp 33–76
25. Dean EJ, Glowinski R (1993) On some finite element methods for the numerical simulation of incompressible viscous flow. In: Gunzburger MD, Nicolaides RA (eds) Incompressible computational fluid dynamics. Cambridge University Press, Cambridge, pp 17–65
26. Del Pin F, Idelsohn SR, O ate E, Aubry R (2005) The ALE/Lagrangian Particle Finite Element Method: a new approach to computation of free-surface flows and fluid-object interactions. *Comput Fluids* (in press)
27. Edelsbrunner H, M cke EP (1994) Three-dimensional alpha shapes. *ACM Trans Graph* 13(1):43–72
28. Eisenstat SC (1981) Efficient implementation of a class of preconditioned conjugate gradient methods. *SIAM J Sci Stat Comput* 2:1–4
29. Elman HC, Silvester DJ, Wathen AJ (2005) Finite elements and fast iterative solvers. Oxford University Press, New York
30. Franca LP, Farhat C (1995) Bubble functions prompt unusual stabilized finite element methods. *Comput Methods Appl Mech Eng* 123:299–308
31. Frey PJ, George PL (1999) Maillages, applications aux elements finis. Hermes, Paris
32. George PL (ed) (2001) Maillage et adaptation. Lavoisier, Paris (in French)
33. Girault V, Raviart PA (1986) Finite element methods for Navier-Stokes equation. Springer, Berlin Heidelberg New York
34. Gresho PM, Lee RL, Sani RL (1984) Further studies on equal-order interpolation for Navier-Stokes. In: Proceedings of the 5th international symposium on finite elements in flow problems. Austin, Texas
35. Henriksen M, Holmen J (2002) Algebraic splitting for Navier-Stokes equations. *J Comput Phys* 175:438–453
36. Hirt CW, Nichols BD (1981) Volume of fluid (VOF) method for the dynamics of free boundaries. *J Comput Phys* 39:201–225
37. Hughes TJR (2000) The Finite element method, linear static and dynamic finite element analysis. Dover Publishers, New York
38. Hughes TJR, Mallet M (1986) A new finite element formulation for computational fluid dynamics: IV. A discontinuity-capturing operator for multidimensional advective-diffusive systems. *Comput Methods Appl Mech Eng* 58:329–339
39. Idelsohn SR, O ate E, Del Pin F (2004) The Particle finite element method: a powerful tool to solve incompressible flows with free-surfaces and breaking waves. *Int J Numer Methods Eng* 61(7):964–989
40. Johnson AA, Tezduyar TE (1999) Advanced mesh generation and update methods for 3D flow simulations. *Comput Mech* 23:130–143
41. L hner R (2001) Applied CFD Techniques. Wiley, New York
42. L hner R (2004) Projective prediction of pressure increments. *Comm Numer Methods Eng* 21:201–207
43. Mardal KA, Winther R (2004) Uniform preconditioners for the time dependent Stokes problem. *Numer Math* 98(2):305–327
44. Mardal KA, Tai XC, Winther R (2002) A robust finite element method for Darcy-Stokes flow. *SIAM J Numer Anal* 40:1605–1631
45. O ate E, Idelsohn SR, Del Pin F, Aubry R (2004) The particle finite element method: an overview. *Int J Comput Methods* 1(2):267–307
46. Osher S, Fedkiw RP (2002) Level set methods and dynamic implicit surfaces. Springer, Berlin Heidelberg New York
47. Page CC, Saunders MA (1975) Solution of sparse indefinite systems of linear equations. *Siam J Numer Anal* 12(4):617–629
48. Pelletier D, Fortin A, Camarero R (1989) Are FEM solutions of incompressible flows really incompressible? (or how simple flows can cause headaches!). *Int J Numer Methods Fluids* 9:99–112
49. Pichelin E, Coupez T (1999) A Taylor discontinuous Galerkin method for the thermal solution in 3d mold filling. *Comput Methods Appl Mech Eng* 178:153–169
50. Pierre R (1989) Regularization procedures of mixed finite element approximations of the Stokes problem. *Numer Methods Part Diff Eq* 5:241–258
51. Pierre R (1995) Optimal selection of the bubble function in the stabilization of the P1-P1 element for the Stokes problem. *Numer Methods Part Diff Eq* 32(4):1210–1224
52. Pierre R (1995) Simple  $C^0$ -approximations for the computations of incompressible flows. *Comput Methods Appl Mech Eng* 68:205–227
53. Prohl A (1997) Projection and Quasi-Compressibility Methods for solving the incompressible Navier–Stokes Equations. Teubner, Stuttgart
54. Quarteroni A, Saleri F, Veneziani A (2000) Factorization methods for the numerical approximation of Navier–Stokes equations. *Comput Methods Appl Mech Eng* 188:505–526
55. Roberts J, Thomas J (1991) Mixed and hybrid methods. In: Ciarlet P, Lions JL (eds) Handbook of numerical analysis, vol. 2. North-Holland, Amsterdam, pp 523–640
56. Saad Y (2003) Iterative methods for sparse linear systems. SIAM, 2nd edn. Philadelphia

57. Saleri F, Veneziani A (2005) Pressure correction algebraic splitting methods for the incompressible Navier–Stokes equations. *SIAM J Numer Anal* 43:174–194
58. Soulaïmani A, Fortin M, Dhatt G, Ouellet Y, Bertrand F (1987) Simple continuous pressure elements for two- and three-dimensional incompressible flows. *Comput Methods Appl Mech Eng* 1:47–69
59. Strang G, Fix G (1973) An analysis of the finite element method. Wellesley-Cambridge Press, Wellesley
60. Témam R (1969) Sur l’approximation de la solution des équations de Navier–Stokes par la méthode des pas fractionnaires. *Arch Rat Mech Anal* 32:135–153
61. Tezduyar TE, Liou J, Ganjoo DJ (1990) Incompressible flow computations based on the vorticity-stream function and velocity-pressure formulations. *Comput Struc* 35:445–472
62. Thomée V (1984) Galerkin finite element methods for parabolic problems. Springer, Berlin Heidelberg New York
63. Thompson JF, Bharat S, Weatherill N (eds) (1999) Handbook of grid generation. CRC Press, West Palm Beach
64. Timmermans LJP, Mineev PD, Van De Vosse FN (1996) An approximate projection scheme for incompressible flow using spectral elements. *Int J Numer Methods Fluids* 22:673–688
65. Turek S (1999) Efficient solvers for incompressible flow problems—an algorithmic and computational approach. Springer, Berlin Heidelberg New York
66. Verfürth R (1984) Error estimates for a mixed finite element approximation of the Stokes equations. *RAIRO Anal Numer* 18:175–182